UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/771,143	01/26/2001	Christopher Crim	CLARP027/P2616	6194

22434 7590 03/06/2008
BEYER WEAVER LLP
P.O. BOX 70250
OAKLAND, CA 94612-0250

EXAMINER	
PHAM, HUNG Q	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
03/06/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/771,143
Filing Date: January 26, 2001
Appellant(s): CRIM ET AL.

RAMIN MAHBOUBIAN
REGISTRATION No. 44,890
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 10/22/2007 appealing from the Office action mailed 05/25/2006 and the request for clarification from Appeal Center 01/24/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

Claims 1-10, 16-37, 44 and 48-50 had been canceled before the Final Office Action 05/18/2005.

After the Final Office Action 05/18/2005, Notice of Appeal was filed on 10/25/2006. Claims 43, 45-47 and 51-58 were canceled in the Amendment filed on 06/07/2007.

Claims 11-15 and 38-42 have been rejected and are being appealed before the Board.

(4) Status of Amendments After Final

In the Amendment filed on 06/07/2007, the appellant canceled claims 43, 45-47 and 51-58 and appealed claims 11-15 and 38-42.

The Amendment filed on 06/07/2007 has been entered.

The appellant's statement of the status of amendments after Final Rejection and Notice of Appeal contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

According to the entered amendment after Final and Notice of Appeal, claims 11-15 and 38-42 are appealed. The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

WITHDRAWN REJECTIONS

The following grounds of rejection are not presented for review on appeal because they have been withdrawn by the examiner.

- The rejection of claims 11-15, 38-42 under 35 U.S.C. § 101;
- The rejection of claims 11 and 38 under 35 U.S.C. § 112, first paragraph.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,236,996 B1	BAPAT ET AL.	05-22-2001

"Fundamentals of Database System", ELMASRI ET AL., ISBN 0-8053-1755-4, Copyright 2000, Page 718.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 11 and 38 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 11 and 38 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01 (As recited in claim 11, a request is received to perform at least one operation on a plurality of records, and evaluating calculation expression for each of said plurality of records. However, evaluating as recited from lines 23-29 is performed only for a first record. The omitted step is identifying and evaluating the next records as disclosed at FIG. 10).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

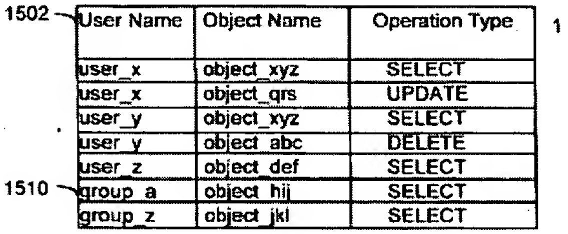
This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 11-15 and 38-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bapat et al. [USP 6,236,996 B1] in view of Elmasri et al. [Fundamentals of Database System].

Regarding claims 11 and 38, Bapat teaches a method and program for controlling managed objects. The method comprising:

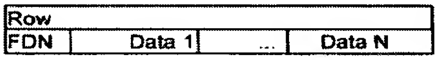
defining a calculation expression, wherein said calculation expression is a variable expression defined based on at least one field of data used in a plurality of records stored in said database (As shown in FIG. 14, tables 310 and 320 as in FIG. 11A are stored in a conventional DBMS 280 (Col. 25, lines 49-50). Rows 311, 312, 321, 322 of the tables 310, 320 contain management information for managed objects (Col. 25, lines 60-61). The FDN operates as the primary key to the data stored in the table and to determine which managed objects that a particular user is permitted to access or modify (Col. 19, lines 36-40). Access control for a particular user on a particular managed object is defined by a permissions table as shown below (Col. 26, lines 10-12).

Granted Permissions Table for Table 1



User Name	Object Name	Operation Type
user_x	object_xyz	SELECT
user_x	object_qrs	UPDATE
user_y	object_xyz	SELECT
user_y	object_abc	DELETE
user_z	object_def	SELECT
group_a	object_hij	SELECT
group_z	object_jkl	SELECT

A permission entry 1502 is tuple having three fields, user name, object name, and operation type. The object name, preferably, is the FDN or Full Distinguish Name for a managed object (Col. 26, Lines 28-33). Referring to FIG. 11A as shown below, each row in the database tables includes a field called the Fully Distinguished Name or FDN of a managed object followed by columns of data. For example, an FDN can look like **/systemid="sys1"/owner="accompany"/devicetype="router"** (Col. 19, Lines 24-35).



Row			
FDN	Data 1	...	Data N

As seen, each row of the Granted Permissions Table is defined by a meaningful combination of variable characters or ***variable expression*** to specify a record access right for a user, wherein each row in the Granted Permissions explicitly defines an access right of a user to a record in the database with its Fully Distinguished Name as a key is equal to the specified Fully Distinguished Name in the Granted Permissions Table. For example, based on the first row of the Granted Permissions Table, a User Name = user_x has Operation Type = delete on any record that has Object Name = object_xyz. Thus, each row expression in the Granted Permissions Table is a ***calculation expression*** with a plurality of implied EQUAL OPERATOR, and is evaluated by the FDN field of the record to determine the access right) and

calculation expression can be evaluated at least partly based on said at least one field of data used in said plurality of records (Col. 28, Lines 1-3, the Grant table is checked to see if user has specific granted items, e.g., FDN, and as discussed above, FDN is *at least one field of data used in said plurality of records* of FIG. 11A),

wherein said at least one filed of data is a variable which may have different values for each of said plurality of records (FIG. 10, tables 310 and 320, FDN field is *a variable which may have different values for each of said plurality of records*),

thereby allowing access to each individual record of said plurality of record to be selectively controlled based on at least one value of said at least one field of data stored for each of said plurality of records of said database (As disclosed by Bapat, the rows of tables 310 and 312 contain management information for managed objects (Bapat, Col. 25 Lines 60-61). Access control, procedure is initiated whenever an SQL command is received. The access control procedure uses permission table to determine whether to grant or deny access to management information stored in DBMS (Bapat, Col. 25 Line 65-Col. 26- Line 3). As shown in FIG. 16A, a user access request to access management information stored in a desire table is intercepted to invoke the access control procedure (Bapat, Col. 29 Lines 31-36). The access control procedure uses the set of access rights stored in the permissions table to determine which rows of data specified by the intercepted query are accessible by the user (Bapat, Col. 29 Lines 39-43). To enforce access control, object name in the form of FDN and user name is used to check against the permissions table to determine whether to grant or deny access (Bapat, Col. 19 Lines 35-40 and Col. 27 Line 45-Col. 28 Line 26). The access control procedure accesses the management information stored in requested rows for which access is permitted by the user (Bapat, Col. 29 Lines 44-47)) and

wherein expression defines access privileges of said one or more users with respect to at least one operation that may be requested to be performed by said one or more users on said plurality of records of said database (FIG. 15 A and B).

When a user 300 issues an SQL command to access the DBMS 280 (Col. 22, lines 24-26, Col. 25, lines 65-67) for the status of all routers in the network or for information about a specified list of managed objects (Col. 28, lines 27-30) with an operation as specified in FIG. 15A as ***receiving a request to perform said at least one operation on said plurality of records of said database, said request being identified as a request made by said one or more users associated with*** user name.

Access Control is enforced by ***evaluating*** user name, object name and operation type as ***said calculation expression for said each of said plurality of records, based on said at least one field of data***, e.g., FDN field, ***when said request has been received***, e.g., SQL command to access management information in DBMS,

wherein said evaluating comprises:

(a) determining at least one value for said at least one field of data stored for a first record of said plurality of records (As disclosed by Bapat, the FDN operates as the primary key to the data stored in the table and to determine which managed objects that a particular user is permitted to access or modify (Col. 19, lines 36-40). As seen, FDN as *value for said at least one field of data stored for a first record of said plurality of records* as in FIG. 11A is determined),

(b) using said at least one value as input to said calculation expression to evaluate said calculation expression for said first record (As disclosed by Bapat, the Grant table is checked to see if user has specific granted items (Col. 28, Lines 1-3). This technique implies FDN is used as *input to* a particular row in Grant table as *calculation expression to evaluate said calculation expression for said first record*),

(c) determining a first result for said calculation expression based on said evaluation of said calculation expression for said first record, wherein said first result effectively indicates whether to grant access to said first record (access is granted if a match occurred (Col. 28, Lines 1-3). As seen, granting access as *a first result* is determined, *wherein said first result effectively indicates whether to grant access to said first record*).

The missing of Bapat is the step of *identifying a password that is associated with one or more users of said database* and Bapat does not teach explicitly that each row of the Permission table is defined for the *identified password*.

As suggested by Bapat, to read the data in a table named "table 1" for a managed object whose FDN is equal to "/a/b/c", an authorized user named "Max" would use the SQL command "SELECT, FROM, WHERE" (Bapat, Col. 20 Lines 28-32).

The step of *identifying a password that is associated with one or more users of said database* is a conventional authorizing technique and taught by Elmasri (Elmasri, page 718).

The Bapat teaching of user authorizing implies the use of a conventional password as taught by Elmasri for protecting access. The defined *calculation expression* in the permissions table for an authorized user implies that user is authorized by a conventional authorizing technique such as user password.

It would have been obvious for one of ordinary skill in the art at the time the invention was made to use the step of identifying a password as taught by Elmasri with the Bapat teaching in order to secure and protect data from misuse and intruders.

Regarding claims 12 and 39, and Bapat and Elmasri, in combination, teach all of the claimed subject matter as discussed above with respect to claims 11 and 38, Bapat further discloses *at least one operation can be a browse, an edit, or a delete operation* (FIG. 15A and B).

Regarding claims 13 and 40, Bapat and Elmasri, in combination, teach all of the claimed subject matter as discussed above with respect to claims 11 and 38, Bapat further discloses *calculation expression is not explicitly defined for said at least one operation but said calculation expression is* *one that has been defined for another operation which has been considered as a related operation to said at least one operation* (FIG. 15A).

Regarding claims 14 and 41, Bapat and Elmasri, in combination, teach all of the claimed subject matter as discussed above with respect to claims 11 and 38, Bapat further discloses *said calculation expression can be evaluated at least partly based on at least one state variable of said database, wherein said state variable can indicate the condition of an element of said database at a particular time* (As further disclosed by Bapat at Col. 26, Lines 55-57 and 60-63, by convention, the permissions tables use a special object name value, such as a database NULL value to represent "all objects". For a system with 5,000 managed objects, only one entry is required (Col. 27, Lines 30-36). GRANT TABLE: (U1, NULL, Op1). Thus, by using NULL variable, the *calculation expression* (U1, NULL, Op1) *can be evaluated based on a state variable of a database*, e.g., NULL indicates 5,000 records, and the number of record is the condition of database at that particular time, because the number of records in the database can be changed overtime, e.g., by deleting or inserting).

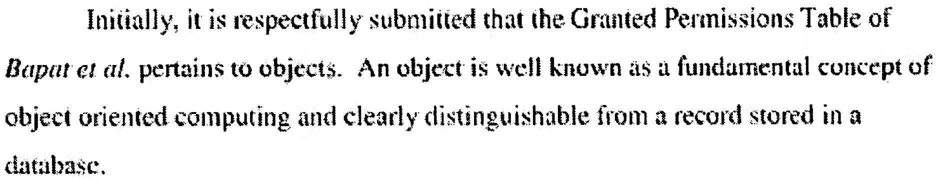
Regarding claims 15 and 42, Bapat and Elmasri, in combination, teach all of the claimed subject matter as discussed above with respect to claims 14 and 38, Bapat further discloses the step of *granting temporary or limited access to said at least one record to allow said evaluating of said calculation expression* (FIG. 15A).

(10) Response to Argument

<u>Response to appellant's arguments with respect to rejection under 35 U.S.C. § 103</u>

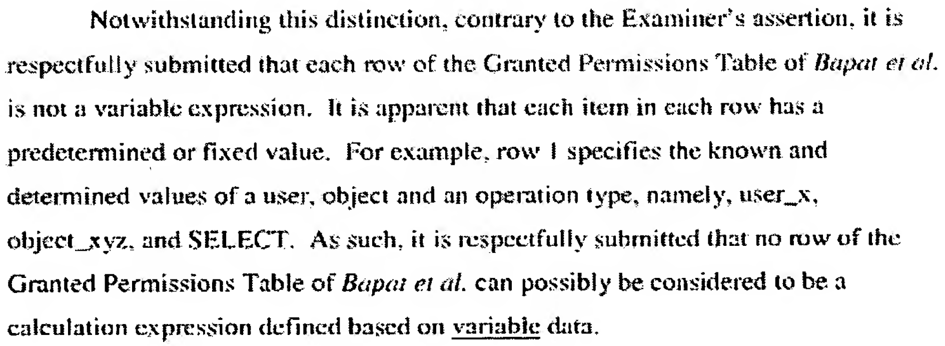
A. The appellant asserted that *defining a calculation expression as a variable expression defined based on a field of data used in records stored in a database, wherein the calculation expression can be evaluated based on the field of data, thereby allowing access to each individual record of the database to be selectively controlled based on a value of a field of data stored for each of the records of the database*[1] (Section 7.1.1 of page 9) was not taught by Bapat. To support for this assertion, the appellant further argued that:

1. (Argument page 10)



Initially, it is respectfully submitted that the Granted Permissions Table of *Bapat et al.* pertains to objects. An object is well known as a fundamental concept of object oriented computing and clearly distinguishable from a record stored in a database.

2. (Argument at page 10)



Notwithstanding this distinction, contrary to the Examiner's assertion, it is respectfully submitted that each row of the Granted Permissions Table of *Bapat et al.* is not a variable expression. It is apparent that each item in each row has a predetermined or fixed value. For example, row 1 specifies the known and determined values of a user, object and an operation type, namely, user_x, object_xyz, and SELECT. As such, it is respectfully submitted that no row of the Granted Permissions Table of *Bapat et al.* can possibly be considered to be a calculation expression defined based on <u>variable</u> data.

[1] Throughout the Examiner Answer, the italic Times New Roman font is used to specify appellant's arguments. The bolded italic Times New Roman font is used to specify the limitation as recited in the claims.

3. (Argument at pages 10 and 11)

Furthermore, it is respectfully submitted that no row of the Granted Permissions Table of *Bapat et al.* defines an expression based on a field of data used in multiple records stored in a database.

In contrast, the claimed invention recites a variable expression that is defined based on a field of data used in a plurality of records, and therefore determining whether to grant access to a particular record is dependent on the actual data stored in that field for that particular record.

4. (Argument at page 11)

Still further, it is respectfully submitted that no row of the Granted Permissions Table of *Bapat et al.* can be used to selectively control access to multiple records. Again, notwithstanding the fact that *Bapat et al.* does not pertain to records of a database and assuming purely for the sake of argument that each row of the Permissions Table of *Bapat et al.* somehow defines a "meaningful combination of variable characters or variable expression" pursuant to the Examiner's assertion, it is apparent the no single row of the Permissions Table of *Bapat et al.* can be evaluated for multiple objects. In other words, even assuming that each row of the Permissions Table of *Bapar et al* is some kind of an expression, it is apparent that this type of expression cannot be evaluated multiple times in order to determine access to multiple entities of a database, regardless of whether these entities are objects in a distributed environment or records stored in a database.

5. (Argument at page 11)

> It should be noted that the Granted Permissions Table of *Bapat et al.* taken as a whole does not teach a calculation expression as a variable expression defined based on a variable field of data used in multiple records stored in a database. *Bapat et al.* teaches a Granted Permissions Table with fixed terms. It is respectfully submitted that those skilled in the art readily appreciate the distinction between a table of fixed terms and a variable expression defined based on one or more variables. In addition, the Granted Permissions Table of *Bapat et al.* does not define a variable expression defined based on a variable field of data used in records stored in a database. Rather, the Granted Permissions Table of *Bapat et al.* is an external table that explicitly specifies access rights of individual users to individual objects.

6. (Argument at pages 11 and 12)

> Accordingly, it is respectfully submitted that the Permissions Table of *Bapat et al.* does not define a variable expression that can be evaluated based on a field of data stored in multiple records (or even objects).
>
> In fact, *Bapat et al.* teaches away from defining a single expression that can be evaluated to define access for multiple records as it teaches providing both a Granted Permission Table (Figure 15A) and a Denied Permissions Table (Figure 15B) in order to provide a comprehensive approach to the general problem of controlling access to objects.

The examiner respectfully disagrees.

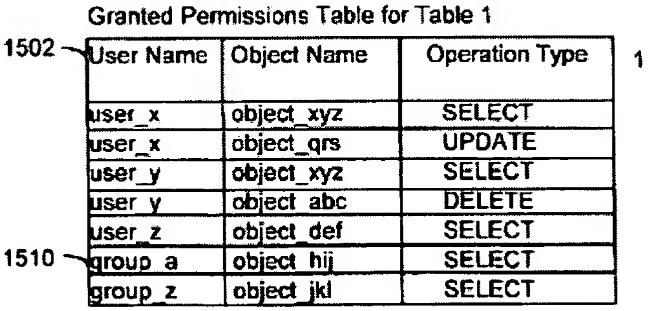
1. Although, the managed objects are implemented by object oriented programming (Bapat, Col. 5 Lines 6-17), however, the managed objects as disclosed by Bapat represent manageable devices in a network (Bapat, Col. 1 Lines 49-56). The permissions table as taught by Bapat pertains to records stored in a database. As disclosed by Bapat at Col. 3 Lines 32-40, access control procedure limits access to the management information stored in the database

tables using permissions table. A permission table defines a subset of rows in the database tables that are accessible to at least one of the users.

The Bapat teaching as discussed clearly indicates the permissions table pertaining to records stored in a database and access to the management information stored in the database tables is controlled by permissions table, e.g., a subset of rows (records) in the database tables are accessible to at least one of the users using permission table.

2. In response to appellant's argument that the references fail to show certain features of appellant's invention, it is noted that the features upon which appellant relies (i.e., *calculation expression defined based on variable data*) are not recited in the rejected claim(s)[2]. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Therefore, this argument does not warrant consideration.

3. As disclosed by Bapat in FIG. 15A, a Granted Permissions Table is copied as below:



Granted Permissions Table for Table 1

User Name	Object Name	Operation Type
user_x	object_xyz	SELECT
user_x	object_qrs	UPDATE
user_y	object_xyz	SELECT
user_y	object_abc	DELETE
user_z	object_def	SELECT
group_a	object_hij	SELECT
group_z	object_jkl	SELECT

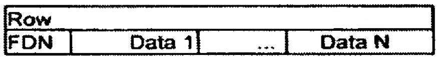
FIG. 15A

[2] As recited in claim 11, *calculation expression... defined based on at least one field of data.*

A permission entry 1502 is a tuple having three fields, user name, object name, and operation type. The object name, preferably, is the FDN or Full Distinguish Name for a managed object (Col. 26, Lines 28-33). For example, an FDN can look like **/systemid="sys1"/owner="accompany"/devicetype="router"** (Col. 19, Lines 24-35).

Permissions table could be updated when a user or object in the permissions table is deleted or added (Bapat, Col. 30 Lines 38-44).

As disclosed by Bapat in FIGS. 10 and 11A, the DBMS stores tables of information, e.g., tables 310 and 320. Each row in the database tables includes a field called the "fully distinguished name" (FDN) followed by columns of data, e.g., data 1... data N, of a managed object (Bapat, Col. 19 Lines 18-27). Referring to FIG. 11A as shown below, each row in the database tables includes a field called the Fully Distinguished Name or FDN of a managed object followed by columns of data.



Row			
FDN	Data 1	...	Data N

FIG. 11A

The Bapat teaching indicates that each row of the Granted Permissions Table is defined by a meaningful combination of characters, which are subject to change when a user or object in the permissions table is deleted or added. Thus, each row of the Granted Permissions Table is a ***variable expression***. Each row in the Granted Permissions is ***defined based on*** FDN field of database tables containing records as ***at least one field of data used in a plurality of records stored in the database.***

4. As disclosed by Bapat, the rows of tables 310 and 312 contain management information for managed objects (Bapat, Col. 25 Lines 60-61). Access control, procedure is initiated whenever an SQL command is received. The access control procedure uses

permission table to determine whether to grant or deny access to management information stored in DBMS (Bapat, Col. 25 Line 65-Col. 26- Line 3). As shown in FIG. 16A, a user access request to access management information stored in a desire table is intercepted to invoke the access control procedure (Bapat, Col. 29 Lines 31-36). The access control procedure uses the set of access rights stored in the permissions table to determine which rows of data specified by the intercepted query are accessible by the user (Bapat, Col. 29 Lines 39-43). To enforce access control, object name in the form of FDN and user name is used to check against the permissions table to determine whether to grant or deny access (Bapat, Col. 19 Lines 35-40 and Col. 27 Line 45-Col. 28 Line 26). The access control procedure accesses the management information stored in requested rows for which access is permitted by the user (Bapat, Col. 29 Lines 44-47).

The Bapat teaching indicates rows of permission table, e.g., set of access right in Granted Permissions Tables as shown in FIG. 15A, are used to selectively control access to multiple records, e.g., rows of data in a desired table specified by user name and FDN in the intercepted query. It is apparent that rows of permission table, e.g., set of access right in Granted Permissions Tables as shown in FIG. 15A, are evaluated, e.g. compared the request's user name and FDN with user name and FDN in the permissions table, to determine access to database records, e.g., access to the management information stored in requested rows is determined whether to grant or deny based on the comparison of request's user name and FDN with user name and FDN in the permissions table.

5. In response to appellant's argument that the references fail to show certain features of appellant's invention, it is noted that the features upon which appellant relies (i.e., *a*

variable expression defined based on a variable field of data used in multiple records[3]) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

The examiner respectfully directs the appellant to the answer to argument 3, where the claimed limitation ***variable expression*** is taught by Bapat.

6. In response to appellant's argument that the references fail to show certain features of appellant's invention, it is noted that the features upon which appellant relies (i.e., *defining a single expression...*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

As discussed above with respect answer to arguments 3 and 4, the Bapat teaching of Permissions Table defines a ***variable expression*** that can be evaluated based on FDN as a field of data stored in multiple records.

B. Bapat teaches *defining a calculation expression that can be evaluated based on a state variable of the database* (claim 14).

As argued by appellant at page 13:

[3] See footnote 1.

> However, it is respectfully submitted that representing all objects or operations with a special value such as NULL does not teach or suggest a calculation expression that can be evaluated based on a state variable of a database. In fact, such global representation requires no evaluation for individual records (or even objects) as by definition it provides or denies access to all of them.

The examiner respectfully disagrees.

As further disclosed by Bapat at Col. 26, Lines 55-57 and 60-63, by convention, the permissions tables use a special object name value, such as a database NULL value to represent "all objects". For a system with 5,000 managed objects, only one entry is required (Col. 27, Lines 30-36). GRANT TABLE: (U1, NULL, Op1). Access control, procedure is initiated whenever an SQL command is received. The access control procedure uses permission table to determine whether to grant or deny access to management information stored in DBMS (Bapat, Col. 25 Line 65-Col. 26- Line 3). As shown in FIG. 16A, a user access request to access management information stored in a desire table is intercepted to invoke the access control procedure (Bapat, Col. 29 Lines 31-36). The access control procedure uses the set of access rights stored in the permissions table to determine which rows of data specified by the intercepted query are accessible by the user (Bapat, Col. 29 Lines 39-43). To enforce access control, object name in the form of FDN and user name is used to check against the permissions table to determine whether to grant or deny access (Bapat, Col. 19 Lines 35-40 and Col. 27 Line 45-Col. 28 Line 26).

Thus, the NULL as taught by Bapat represents 5,000 records. The number of record is the condition of database at that particular time due to the fact that the number of records in the database can be changed by deleting and inserting operations. The NULL as taught by Bapat is considered as the claimed limitation ***a state variable of a database***. By using NULL variable, the

calculation expression (U1, NULL, Op1) will be matched against particular user U1 and any FDN in the request. In different words, (U1, NULL, Op1) ***can be evaluated based on a state variable of a database***.

C. Bapat teach *evaluating a calculation expression for a plurality of records based on a field of data stored for each record* as recited in claims 11 and 38.

As argued by appellant at pages 13-14:

> If a row of the Granted Permissions Table of *Bapat et al.* can be considered to be a calculation expression, the Examiner needs to show that *Bapat et al.* teaches evaluating the row for multiple records (or at least multiple objects). Furthermore, checking a permission table to determine whether an entry exists (or does not exist) for a particular record is not the same as evaluating a calculation expression multiple times. Still further, searching a table to find an entry does not teach evaluating a variable expression based on data stored in a particular field of a record in order to determine whether to grant access to that particular record. Again, it should be noted that *Bapat et al.* teaches using both a Granted Permissions Table and a Denied Permissions Table (Figures 15A and 15B). Hence, the methodology of *Bapat et al.* teaches away from evaluating a single calculation expression in order to control access to multiple records stored in a database as it teaches searching not just one but multiple tables in order to control access to objects.

The examiner respectfully disagrees.

As discussed above with respect to answer to argument 3, a row of the Granted Permissions Table is considered to be ***a calculation expression***. To show that Bapat teaches the row is evaluated for multiple records, the examiner respectfully directs the appellant to FIG. 16A of Bapat. As shown in FIG. 16A, a user access request to access management information stored in a desire table is intercepted to invoke the access control procedure (Bapat, Col. 29

Lines 31-36). The access control procedure uses the set of access rights stored in the permissions table to determine which rows of data specified by the intercepted query are accessible by the user (Bapat, Col. 29 Lines 39-43). To enforce access control, object name in the form of FDN and user name is used to check against the permissions table to determine whether to grant or deny access (Bapat, Col. 19 Lines 35-40 and Col. 27 Line 45-Col. 28 Line 26). The access control procedure accesses the management information stored in requested rows for which access is permitted by the user (Bapat, Col. 29 Lines 44-47).

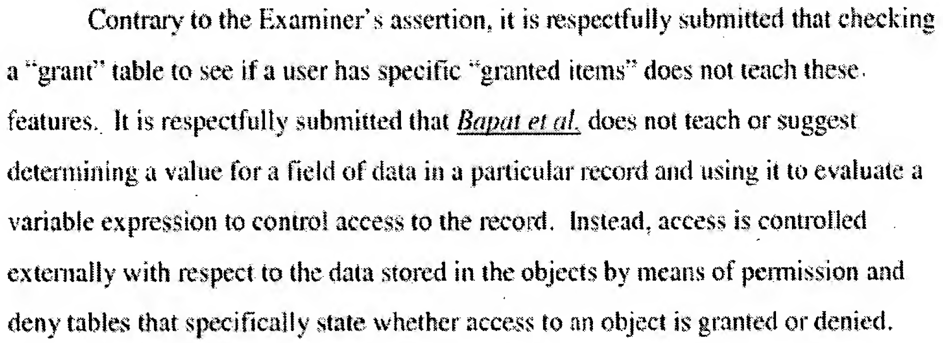
Thus, an access request from a user includes user name and FDN. To enforce access control, the row that contains the requested FDN corresponding to multiple records of that FDN is checked or ***evaluated*** against the requested user name and FDN to determine whether to grant or deny access. The FDN is considered to be ***at least one field of data*** as further recited in the step of "***evaluating the calculation expression***". In short, the Bapat technique indicates the step of evaluating a calculation expression for a plurality of records based on a field of data stored for each record.

In response to appellant's argument that the references fail to show certain features of appellant's invention (i.e., *checking a permission table to determine whether an entry exists (or does not exist) for a particular record is not the same as evaluating a calculation expression multiple times*), it is noted that the features upon which appellant relies (i.e., *evaluating a calculation expression multiple times*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In response to appellant's argument that *Bapat et al. teaches away from evaluating a single calculation expression* by using both a table Granted Permissions Table and a Denied Permissions Table, it is noted that a row in either Granted Permissions Table or Denied Permissions Table is considered to be ***a calculation expression***. The recited ***calculation expression*** is not distinguished over the Bapat expression as discussed above with respect to answer to argument 3.

D. Bapat teaches *determining at least one value for a field of data stored in a record and using it to evaluate a variable expression in order to control access to that record* (claims 11 and 38).

As argued by appellants at page 14:



> Contrary to the Examiner's assertion, it is respectfully submitted that checking a "grant" table to see if a user has specific "granted items" does not teach these features. It is respectfully submitted that <u>*Bapat et al.*</u> does not teach or suggest determining a value for a field of data in a particular record and using it to evaluate a variable expression to control access to the record. Instead, access is controlled externally with respect to the data stored in the objects by means of permission and deny tables that specifically state whether access to an object is granted or denied.

The examiner respectfully disagrees.

As disclosed by Bapat, the access control procedure uses the set of access rights stored in the permissions table to determine which rows of data specified by the intercepted query are accessible by the user (Bapat, Col. 29 Lines 39-43). To enforce access control, FDN is used to determine which managed objects that a particular user is permitted to access or modify (Bapat, Col. 19 Lines 35-40).

The Bapat teaching indicates *at least one value for a field of data stored in a record*, e.g., the FDN value, is *determined* and *used* to check or *evaluate* against a row in the Permissions Table as *variable expression* to control access rows of data specified by the intercepted query.

E. The examiner has established a prima facie case of obviousness and provided a motivation or suggestion for defining a calculation expression for a password.

Appellant's arguments with respect to a prima facie case of obviousness (Sections 7.1.5, 7.1.6 and 7.1.7 on pages 14 and 15) have been fully considered but they are not persuasive.

As discussed in the Final Action 05/25/2006, the missing of Bapat is the step of *identifying a password that is associated with one or more users of said database* and each row of the Permission table is defined for the *identified password*.

As suggested by Bapat, to read the data in a table named "table 1" for a managed object whose FDN is equal to "/a/b/c", an authorized user named "Max" would use the SQL command "SELECT, FROM, WHERE" (Bapat, Col. 20 Lines 28-32).

The step of *identifying a password that is associated with one or more users of said database* is a conventional authorizing technique and taught by Elmasri (Elmasri, page 718).

The Bapat teaching of user authorizing implies the use of a conventional password as taught by Elmasri for protecting access. The defined *calculation expression* in the permissions table for an authorized user implies that user is authorized by a conventional authorizing technique such as user password.

It would have been obvious for one of ordinary skill in the art at the time the invention was made to use the step of identifying a password as taught by Elmasri with the Bapat teaching in order to secure and protect data from misuse and intruders.

Response to appellant's arguments with respect to rejection under 35 U.S.C. § 112, second paragraph

In response to appellant's argument that identifying and evaluating the second record as disclosed at FIG. 10 is not essential, the examiner respectfully points out that in claims 11 and

38, a request is received to perform at least one operation on plurality of records, and evaluating calculation expression for each of plurality of records. However, evaluating as further defined (e.g., lines 23-29 of claim 11) is performed only for a first record. Therefore, the process is being incomplete for omitting essential steps, e.g., identifying and evaluating the next records as disclosed in FIG. 10 of the current invention.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Examiner Hung Pham
AU 2168

Conferees:

MOHAMMAD ALI, Supervisory Patent Examiner, AU 2163

TIM VO, Supervisory Patent Examiner, AU 2168

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100